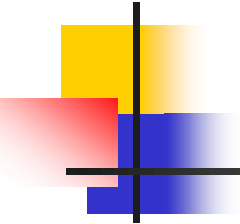




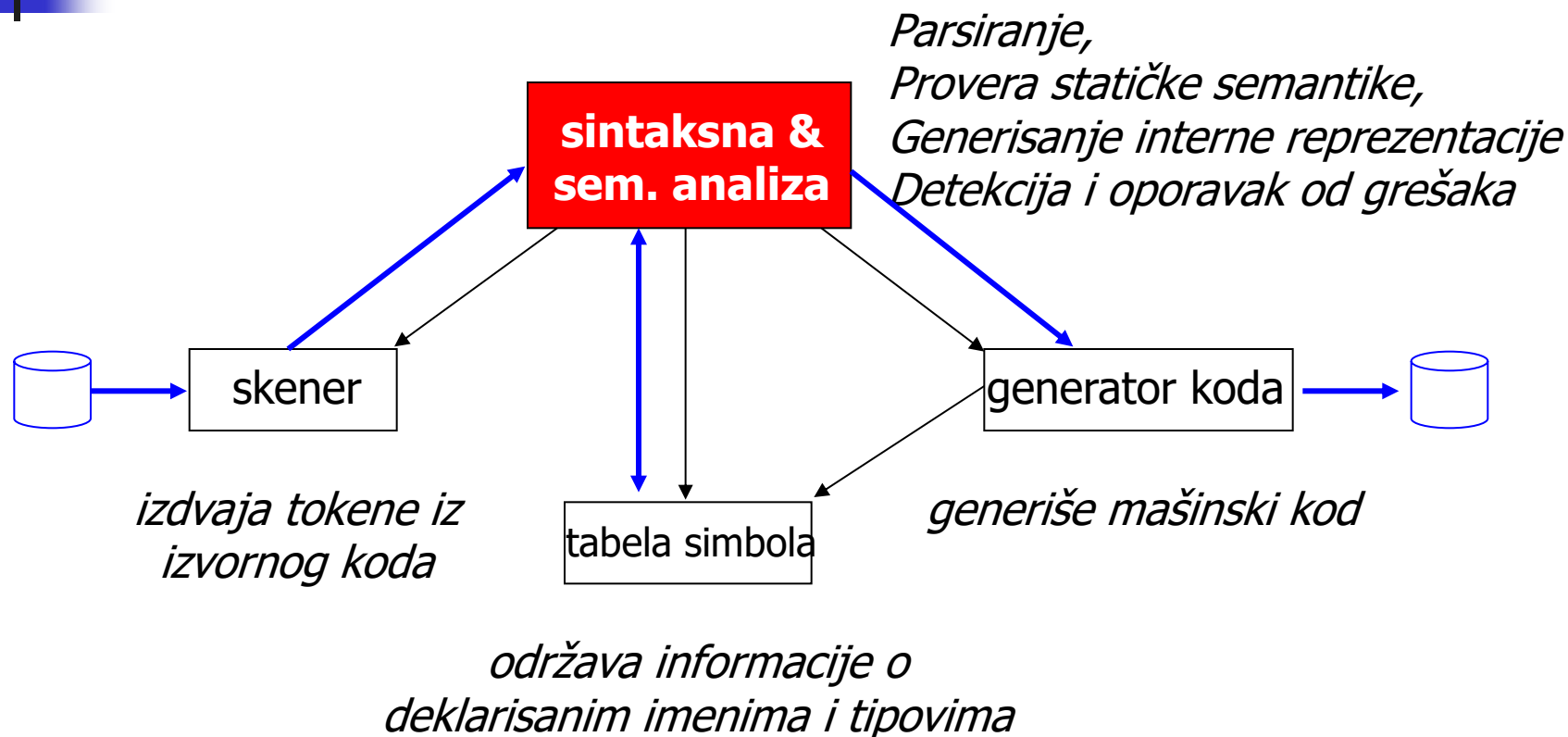
# Programski prevodioci 1

---

## Lekcija – Sintaksna analiza

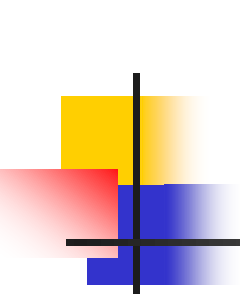
- 
- Uloga sintaksno/semantičke analize
  - Gramatike
  - Višeznačnost gramatika
  - Klasifikacija gramatika

# Uloga sintaksno-semantičke analize



→ Poziva: sintaksno upravljano prevođenje

→ tok podataka

- 
- 
- Uloga sintaksno/semantičke analize
  - Gramatike
  - Višeznačnost gramatika
  - Klasifikacija gramatika



# Bezkontekstna gramatika

- Bezkontekstna gramatika  $G$  opisana je uređenom četvorkom  $(V_N, V_T, P, \langle S \rangle)$  gde:
  - $V_N$  predstavlja skup neterminalnih simbola, npr.  $V_N = \{\langle S \rangle, \langle A \rangle, \langle B \rangle\}$ .
  - $V_T$  predstavlja skup terminalnih simbola, npr.  $V_T = \{a, b, c, d\}$ .  $V = V_T \cup V_N$  predstavlja skup gramatičkih simbola.
  - $P$  predstavlja skup smena (produkcija) oblika  $\langle X \rangle \rightarrow \alpha$ , gde  $\langle X \rangle \in V_N$  a  $\alpha$  na predstavlja proizvoljan niz od nula ili više gramatičkih simbola.
  - $\langle S \rangle \in V_N$  predstavlja startni neterminal.

# BNF Notacija

<i>terminalni simboli</i>	pišu se bez zagrada (ident, +, -)
<i>neterminalni simboli</i>	pišu se u zagradama (<Expr>, <Term>)
<i>strane smene</i>	razdvajaju se sa ::= ili →

## BNF gramatika za aritmetičke izraze

```
<Expr>    → <Sign> <Term>
<Expr>    → <Expr> <Addop> <Term>
<Sign>    → +
<Sign>    → -
<Sign>    → ε
<Addop>   → +
<Addop>   → -
<Term>    → <Factor>
<Term>    → <Term> <Mulop> <Factor>
<Mulop>   → *
<Mulop>   → /
<Factor>  → ident
<Factor>  → number
<Factor>  → ( <Expr> )
```

- Alternative su transformisane u posebne smene
- Ponavljanje se mora izraziti rekurzijom

## Prednosti

- manje meta simbola (bez |, (), [], {})
- Lakše napraviti sint. stablo

## Mane

- više smena

# EBNF Notacija

**Extended Backus-Naur form**  
(BNF + reg. izrazi)

*John Backus:* razvio prvi Fortran compiler  
*Peter Naur:* editovao Algol60 specifikaciju

<i>simbol</i>	<i>značenje</i>	<i>primeri</i>
<b>string</b>	Predstavlja “sam sebe”	"=", "while"
<b>ime</b>	predstavlja T ili NT simbol	ident, Statement
<b>=</b>	razdvaja strane smene	A = b c d .
<b>•</b>	završava smenu	
<b> </b>	Razdvaja alternative	a   b   c    značenje   a or b or c
<b>(...)</b>	grupiše alternative	a ( b   c )    značenje   ab   ac
<b>[...]</b>	opciona deo	[ a ] b    značenje   ab   b
<b>{...}</b>	ponavljajući deo	{ a } b    značenje   b   ab   aab   aaab   ...

## Konvencije

- Terminalni simboli počinju malim slovom (npr. ident)
- Neterminalni simboli počinju velikim slovima (npr. Statement)

# EBNF, Sintaksni dijagrami

## Smene

$\text{Expr} = [ "+" \mid "-" ] \text{Term} \{ ( "+" \mid "-" ) \text{Term} \}.$   
 $\text{Term} = \text{Factor} \{ ( "*" \mid "/" ) \text{Factor} \}.$   
 $\text{Factor} = \text{ident} \mid \text{number} \mid "(" \text{Expr} ")".$

## Terminalni simboli

**Obični TS:**  $"+"$ ,  $"-"$ ,  $"*"$ ,  $"/"$ ,  $"("$ ,  $)"$   
(samo 1 instanca)

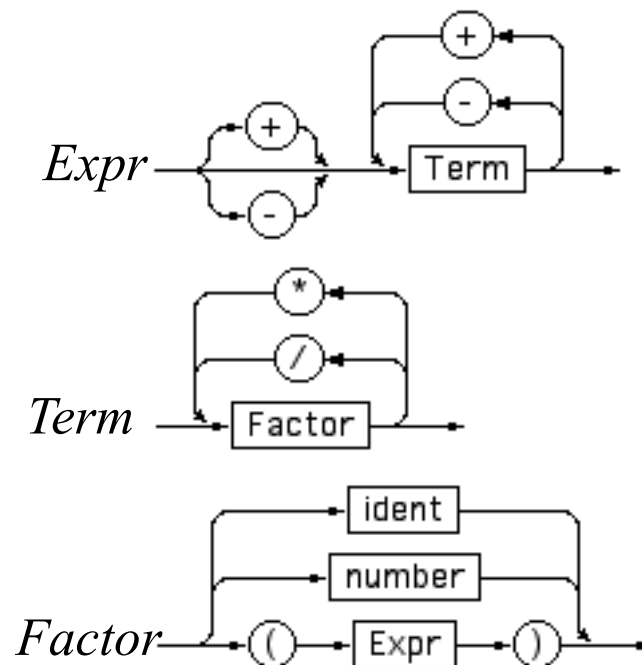
**klase TS:**  $\text{ident}$ ,  $\text{number}$   
(više instanci)

## Neterminalni simboli

$\text{Expr}$ ,  $\text{Term}$ ,  $\text{Factor}$

## Startni simbol

$\text{Expr}$





# Terminologija

- **Alfabet** posmatrane gramatike – skup terminala  $V_T$
- **Sentencijalna forma** – proizvoljan niz gramatičkih simbola
  - oznaka  $\alpha, \beta, \dots$  ili  $ABCD, XYZ, \dots$
- **Sentenca** – sekvenca terminalnih simbola opisanih gramatikom
  - Oznaka  $uvw, \dots$
- **Gramatičko izvođenje**, u oznaci
$$\beta <X> \gamma \Rightarrow \beta \alpha \gamma$$

predstavlja primenu neke gramatičke smene  $<X> \rightarrow \alpha$  da bi se u nekom nizu gramatičkih simbola (takozvanoj sentencijalnoj formi)  $\beta <X> \gamma$  zamenilo pojavljivanje leve strane smene odgovarajućom desnom stranom.



# Terminologija (nastavak)

- $\Rightarrow^+$  izvođenje u jednom ili više koraka

$$\alpha \Rightarrow \beta \Rightarrow \dots \Rightarrow \gamma$$

- $\Rightarrow^*$  naziva se izvođenje u nula ili više koraka.
- **Redukcija** – obrnuto od izvođenja, zamena desne strane smene levom u nekoj sent. formi
- **Jezik** opisan gramatikom  $G$ , u oznaci  $L(G)$  predstavlja skup svih sentenci koje se mogu izvesti iz startnog neterminala  $\langle S \rangle$  u nula ili više koraka izvođenja:

$$L(G) = \{x \mid \langle S \rangle \Rightarrow^* x, x \in V_T^*\}$$



# Primer

Gramatika G1:

1.  $\langle S \rangle \rightarrow a \langle A \rangle c \langle B \rangle$
2.  $\langle S \rangle \rightarrow \langle B \rangle d \langle S \rangle$
3.  $\langle B \rangle \rightarrow a \langle S \rangle c \langle A \rangle$
4.  $\langle B \rangle \rightarrow c \langle A \rangle \langle B \rangle$
5.  $\langle A \rangle \rightarrow \langle B \rangle a \langle B \rangle$
6.  $\langle A \rangle \rightarrow a \langle B \rangle c$
7.  $\langle A \rangle \rightarrow a$
8.  $\langle B \rangle \rightarrow b$

Izvođenje:

$$\begin{array}{ccccccc} \langle S \rangle & \Rightarrow & a & \langle A \rangle & c & \langle B \rangle & \Rightarrow & a & a & c & \langle B \rangle & \Rightarrow & a & a & c & b \\ \uparrow & & & \uparrow & & & & & & \uparrow & & & & & & \\ 1 & & & 7 & & & & & & 8 & & & & & & \end{array}$$

$aacb \in L(G1)$



# Terminologija (nastavak)

- **levo izvođenje (engl. leftmost derivation)** - u svakom koraku izvođenja zamenjivan prvi neterminal s leve strane

$\langle S \rangle \Rightarrow_{lm} a \langle A \rangle c \langle B \rangle \Rightarrow_{lm} a a c \langle B \rangle \Rightarrow_{lm} a a c b$

- **desno izvođenje (engl. rightmost derivation)** - u svakom koraku izvođenja zamenjuje se krajnje desni neterminal u sentencijalnoj formi

$\langle S \rangle \Rightarrow_{rm} a \langle A \rangle c \langle B \rangle \Rightarrow_{rm} a \langle A \rangle c b \Rightarrow_{rm} a a c b$

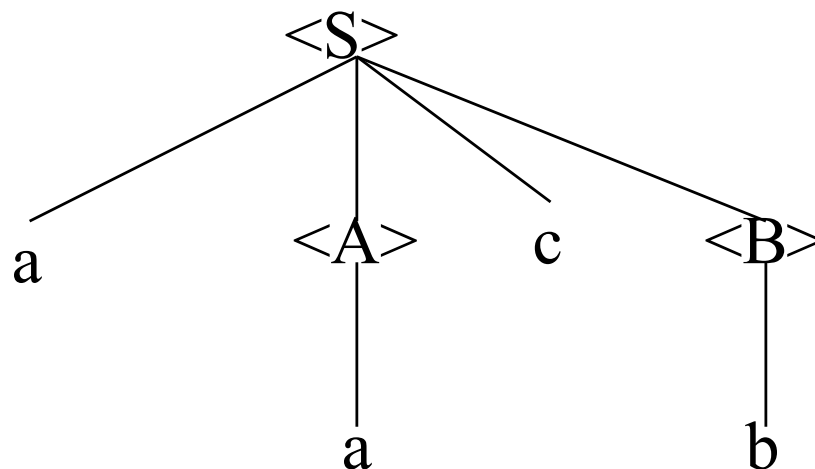
- **Parser** je procedura koja na osnovu date gramatike utvrđuje, za zadatu sekvencu, pripadnost sekvence jeziku opisanom datom gramatikom i njeno izvođenje iz startnog neterminala

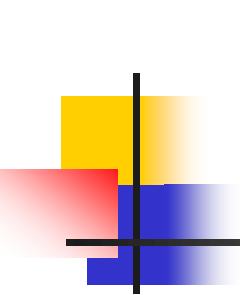
# Terminologija (nastavak)

- **Stablo izvođenja** je grafička predstava izvođenja

$\langle S \rangle \Rightarrow a \langle A \rangle c \langle B \rangle \Rightarrow a a c \langle B \rangle \Rightarrow a a c b$   
                  ↑                  ↑                  ↑  
                  1                  7                  8

- Izvođenje  $\langle X \rangle \rightarrow Y Z$  V se u stablu izvođenja predstavlja na taj način što čvor  $\langle X \rangle$  ima čvorove naslednike Y, Z i V navedenim redosledom..



- 
- 
- Uloga sintaksno/semantičke analize
  - Gramatike
  - Višeznačnost gramatika
  - Klasifikacija gramatika

# Višeznačnost gramatike

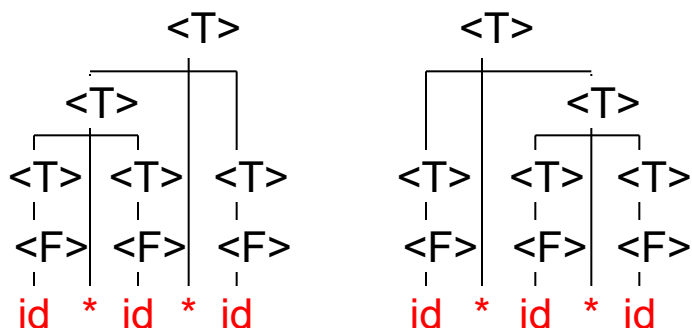
Ako je gramatika višeznačna, za jednu sentencu moguće je odrediti više različitih stabala izvođenja

## Primer

$\langle T \rangle \rightarrow \langle F \rangle \mid \langle T \rangle * \langle T \rangle$   
 $\langle F \rangle \rightarrow \text{id}$

sentenca: id \* id \* id

Za ovu sentencu moguće je konstruisati dva sintaksna stabla:



Ovo je nepoželjno svojstvo pri prevođenju.

# Otklanjanje višeznačnosti

## Primer

$\langle T \rangle \rightarrow \langle F \rangle \mid \langle T \rangle * \langle T \rangle$   
 $\langle F \rangle \rightarrow \text{id}$

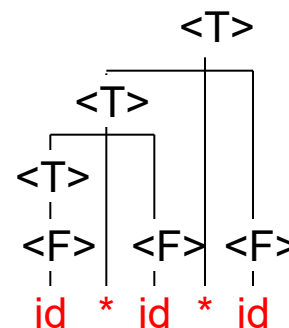
Jezik nije dvosmislen, samo gramatika

## Gramatika se može transformisati u

$\langle T \rangle \rightarrow \langle F \rangle \mid \langle T \rangle * \langle F \rangle$   
 $\langle F \rangle \rightarrow \text{id}$

Tako da  $\langle T \rangle$  ima veći prioritet od  $\langle F \rangle$

Sada postoji samo jedno sintaksno stablo





# Dvosmislenost u prog. jezicima

## Primer: *Dangling Else (viseće Else)*

```
<Stmt> →    if <Expr> then <Stmt>
            |    if <Expr> then <Stmt> else <Stmt>
            |    other Stmts
```

Razmotrimo sentencu:

If  $E_1$  then if  $E_2$  then  $S_1$  else  $S_2$

Ova sentenca ima dva stabla izvođenja.

**Moguće rešenje** (**else** se pridružuje najbližem neuparenom **then**)

```
<Stmt>      →  <Matched>
               |  <Unmatched>
<Matched> →  if <Expr> then <Matched> else <Matched>
               |  other Stmts
<Unmatched> → if <Expr> then <Stmt>
               |  if <Expr> then <Matched> else <Unmatched>
```



## Dvosmislenost u prog. jezicima (2)

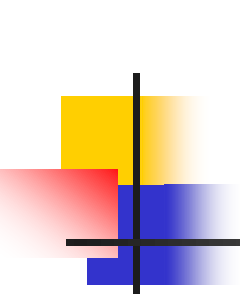
**Primer:** *Dvosmislenost kao posledica preklapanja identifikatora*

`a = f(17)`

Da li je *f* funkcija, ili vektor (u npr. Algolu)?

Da bi se ova dvosmislenost razrešila,  
neophodan je semantički kontekst:

- potrebno je poznavati tip simbola *f*

- 
- Uloga sintaksno/semantičke analize
  - Gramatike
  - Višeznačnost gramatika
  - Klasifikacija gramatika

# Klasifikacija gramatika

Noam Chomsky (1956)

**Gramatike su skupovi smena opšteg oblika  $\alpha \rightarrow \beta$ .**

**klasa 0 Neograničene gramatike** ( $\alpha$  i  $\beta$  proizvoljni)

npr:  $\langle A \rangle \rightarrow a \langle A \rangle b \mid \langle B \rangle c \langle B \rangle$

$a \langle B \rangle c \rightarrow d \quad \langle A \rangle \Rightarrow a \langle A \rangle b \Rightarrow a \langle B \rangle c \langle B \rangle b \Rightarrow d \langle B \rangle b \Rightarrow bbb$

$d \langle B \rangle \rightarrow bb$

Prepoznaju se Tjuringovim mašinama

**klasa 1 Kontekstne gramatike** ( $|\alpha| \leq |\beta|$ )

npr:  $a \langle A \rangle \rightarrow a b c$

Prepoznaju se linearno ograničenim automatima

**klasa 2 Context-free grammars** ( $\alpha = NT$ )

npr:  $\langle A \rangle \rightarrow a b c$ .

Prepoznaju se potisnim automatima

**klasa 3 Regularne gramatike** ( $\alpha = NT, \beta = \varepsilon \mid T \mid NT$ )

npr:  $\langle A \rangle \rightarrow \varepsilon \mid b \langle B \rangle$ .

Prepoznaju se konačnim automatima

Samo ove dve klase  
su relevantne za  
konstrukciju  
prevodilaca